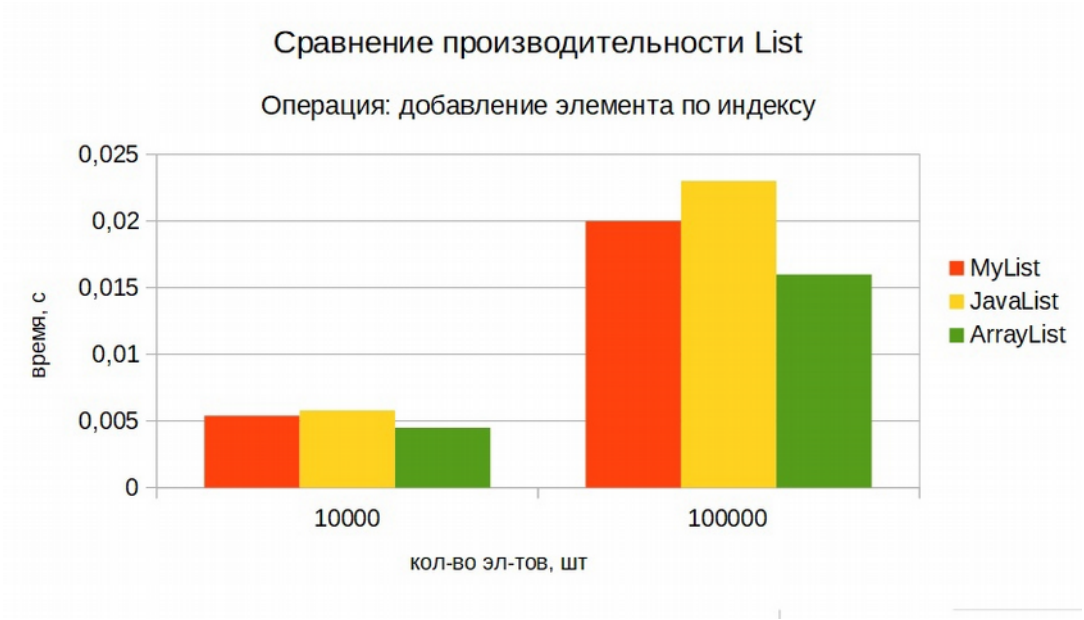
Сравнение производительности коллекций для основных операций

1. ArrayList, LinkedList, MyLinkedList



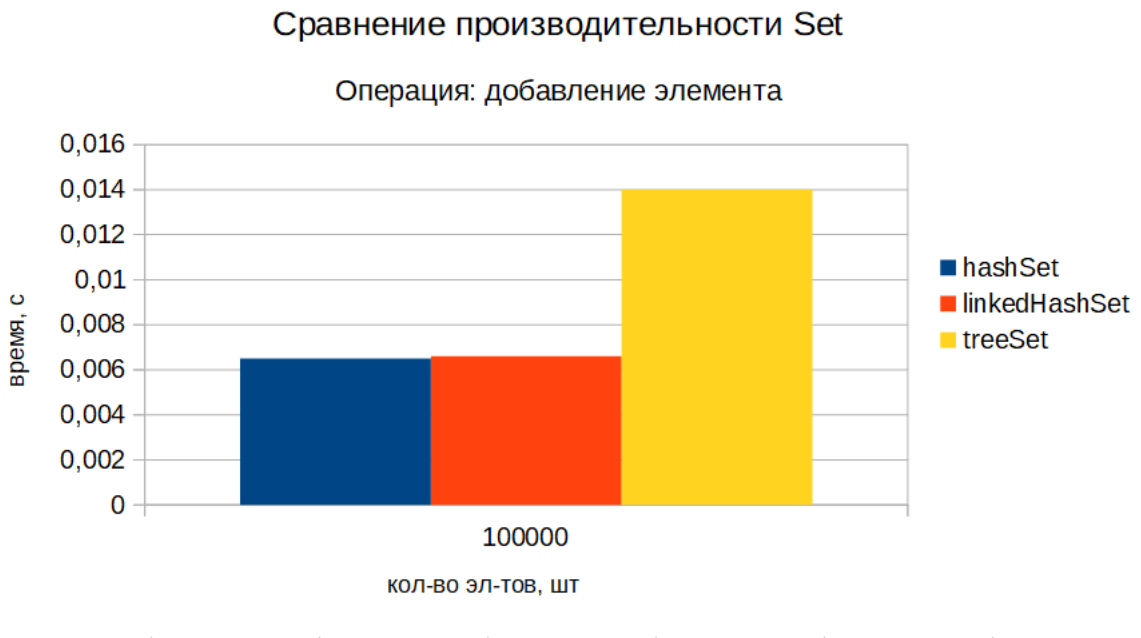
Исходя из результатов видно, что ArrayList наиболее эффективен для всех операций. Однако в проведенных тестах индекс генерировался случайно и данных было много, поэтому могли происходить разные ситуации, которые значительно влияют на эффективность, например:

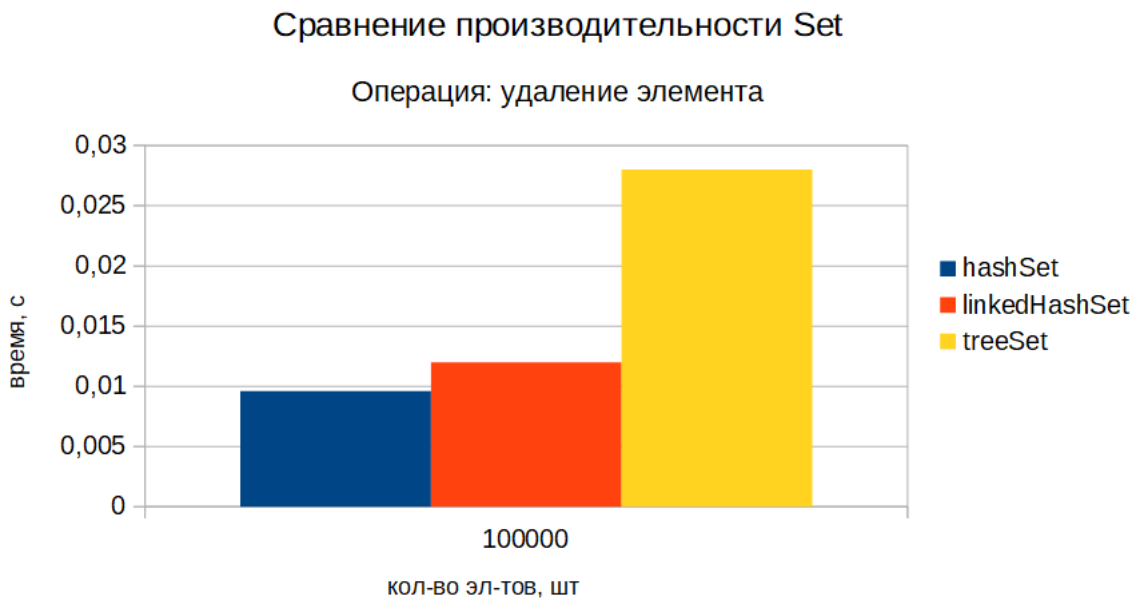
1) если значения индексов i генерировались ближе к значению size/2, то тогда LinkedList будет неэффективен и лучше использовать ArrayList.

2) если значения индексов i генерировались ближе к началу в ArrayList, то тогда он будет не эффективен при вставке и удалении, так как происходит копирование во временный массив и обратно. Поэтому лучше использовать LinkedList.

Использование памяти также отличается у ArrayList и LinkedList. Каждый элемент LinkedList имеет больше расходов, поскольку хранятся указатели на следующий и предыдущий элементы. ArrayList не имеет этих расходов, но он занимает столько памяти, сколько выделено для емкости, независимо от того, были ли элементы добавлены на самом деле.

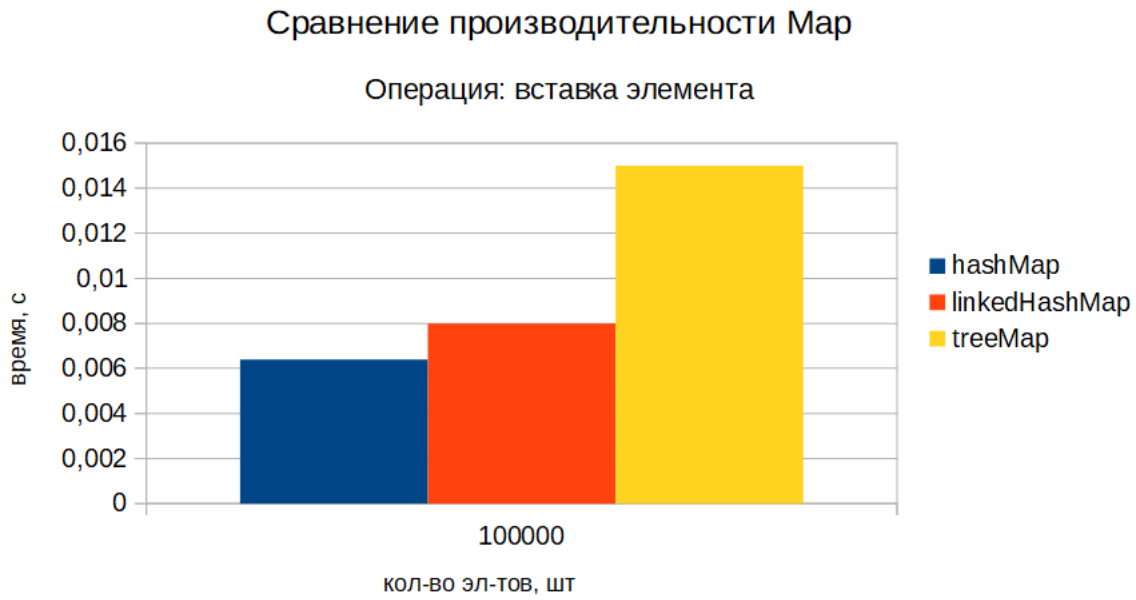
2. HashSet, LinkedHashSet, TreeSet

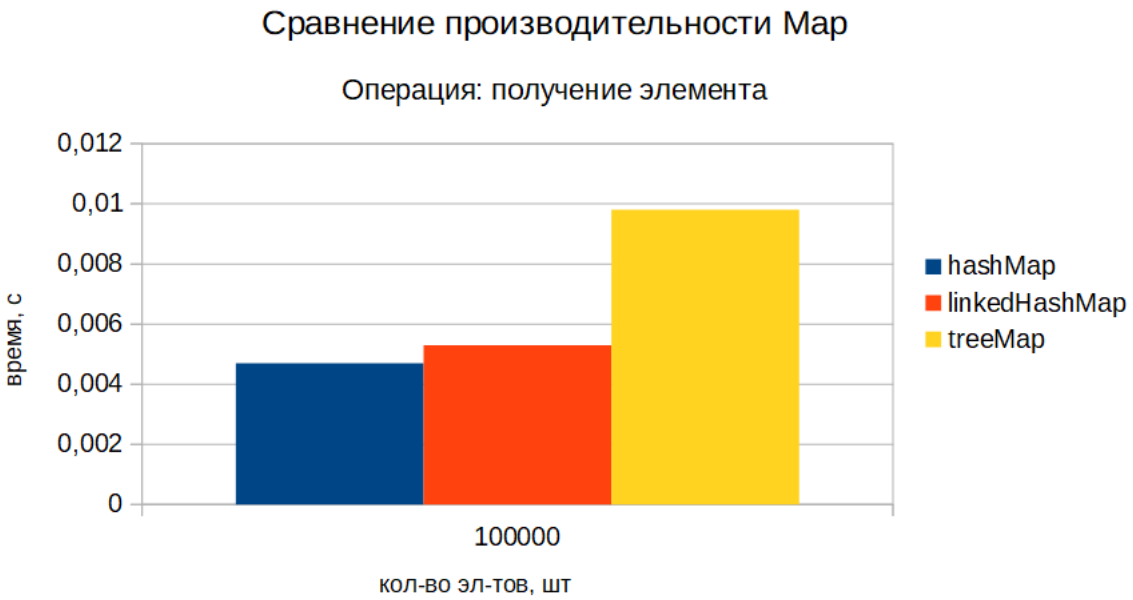


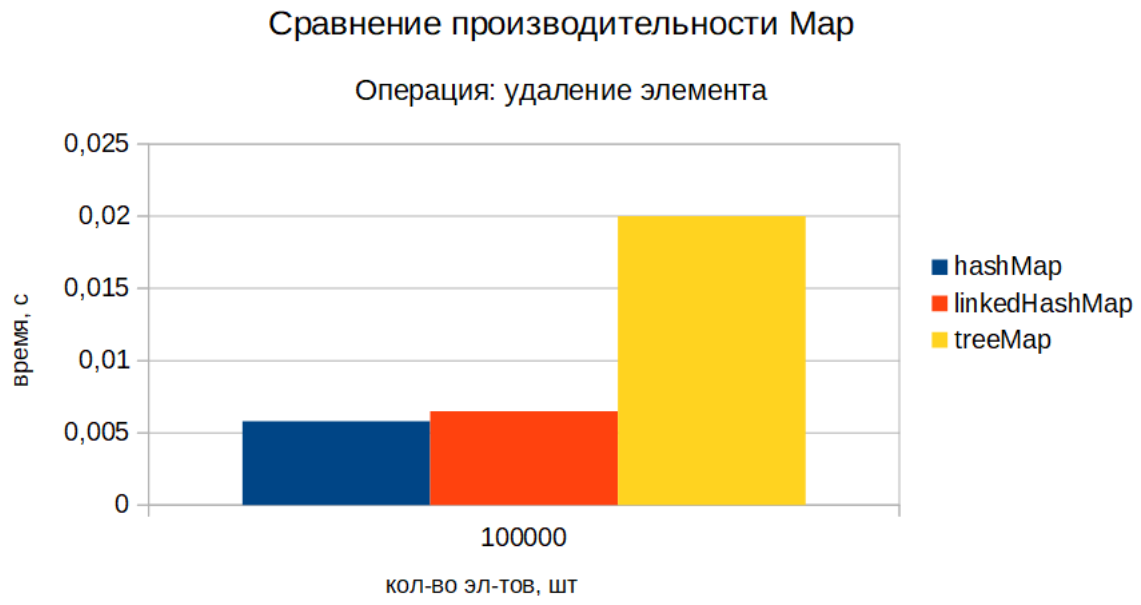


Set представляет собой множество без дублирующих элементов. Если нам нужна быстрая коллекция, используем HashSet, если отсортированная коллекция, то TreeSet, если нужна коллекция, которая может хранить порядок вставки, используем LinkedHashSet.

3. HashMap, LinkedHashMap, TreeMap







Map – структура, которая хранит пару (ключ, значение). Выбор аналогично структурам Set.